COL216 Assignment-3

2021CS10083 Vidushi Maheshwari
2021CS10562 Raghav Ajmera

May 11, 2023

§1 Design Decisions

§1.1 Inclusivity

- Inorder to maintain the inclusivity (i.e. if a block is present in L1 cache, it must be present in L2 cache as well), everytime we removed a block from L1 cache, we evicted the block from L2 cache also. This ensures that the number of L2 write misses is 0.

§1.2 Read Command

- In case of L1 read miss and L2 read hit, if the corresponding block in L1 cache where the data from L2 cache is supposed to be read to is a valid and dirty bit, the data stored in the block of L1 cache is first written back to L2 cache and then the read block from L2 cache is imported to L1 cache and the dirty bit of the block is set to be 0.

- Similar design decision was taken in case of L2 read miss.

§1.3 Write Command

- We enforced the write-back and write-allocate policies.

- In case of L1 write miss, we write only to L1 cache and not L2 cache or Memory.

- L2 writes is counted only in case of write-back from L1 cache to L2 cache.

§1.4 LRUs

- We maintained a 2-d vector of size equal to

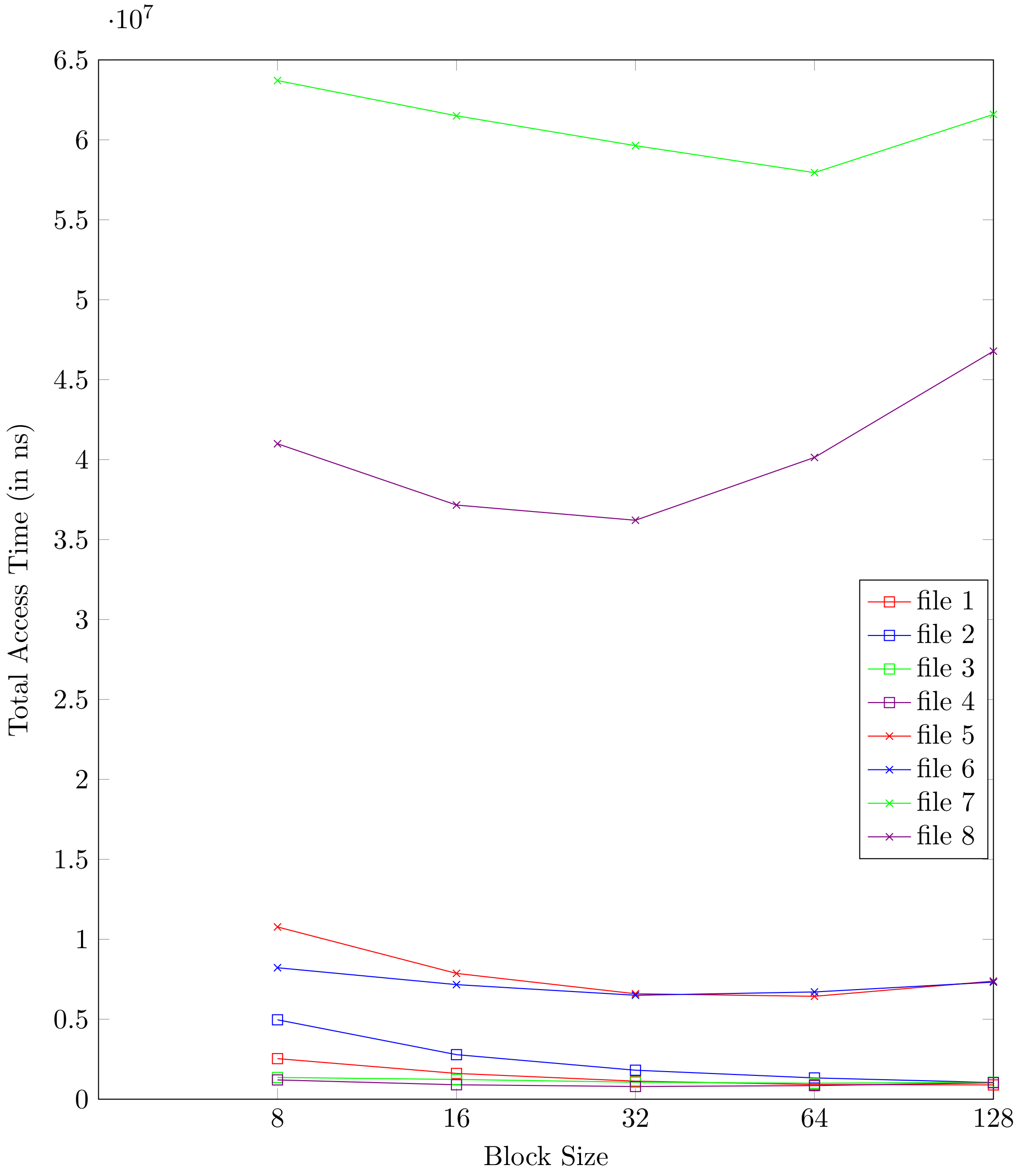
$$number\ of\ rows * associativity\ of\ the\ cache$$

- 0 is the value of most recently used block and in case of completely filled line, *Associativity - 1* is the value of least recently used block.

- In case of update, we increment all values by 1 and set the value of given index as 0. This significantly brought down the number of write and read misses.

§2 Graph for Total Access Time vs Block Size

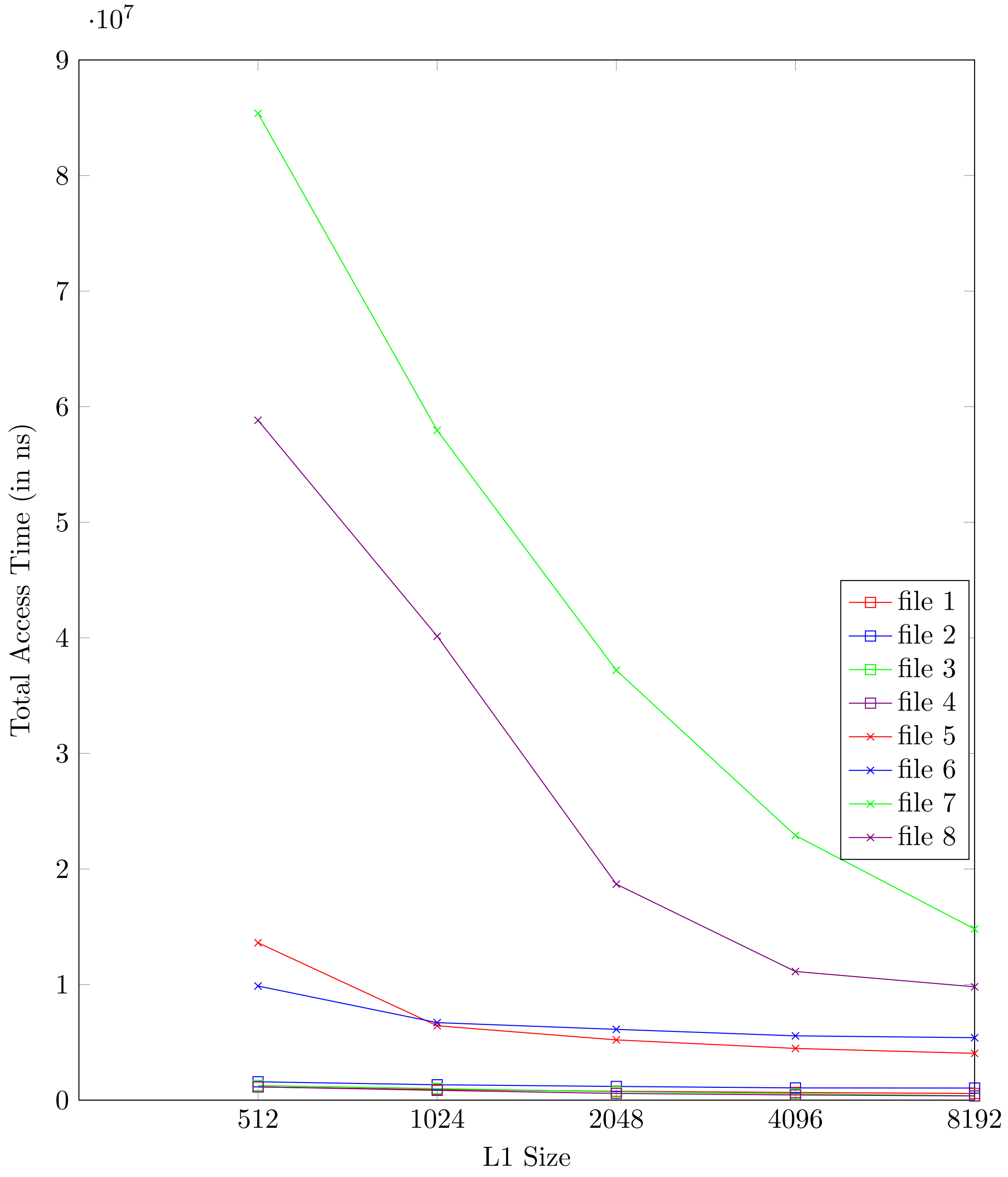
- L1 size = 1024, L1 Associativity = 2, L2 size = 65536, L2 Associativity = 8



Block Size	8	16	32	64	128
file 1	2537100	1611560	1127900	915740	892360
file 2	4965420	2785660	1811820	1330500	1044380
file 3	1359520	1234560	1062900	998540	1061360
file 4	1204560	901560	792920	848880	1030420
file 5	10772600	7866320	6594200	6433860	7386000
file 6	8220940	7164900	6500640	6706500	7321600
file 7	63707360	61503140	59634940	57948460	61592220
file 8	40992140	37158140	36207620	40135320	46784200

§3 Graph for Total Access Time vs L1 Size

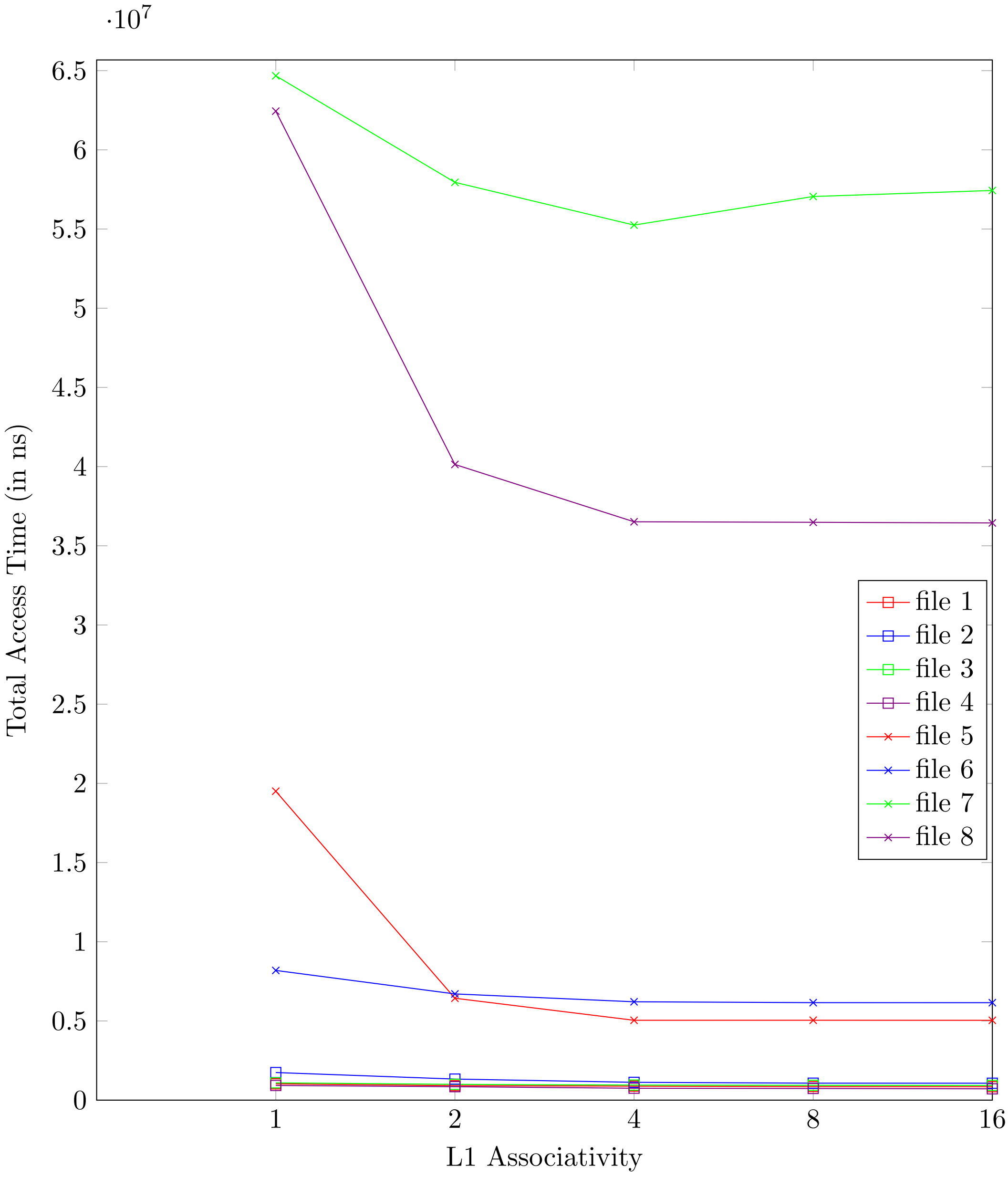
- Block size = 64, L1 Associativity = 2, L2 size = 65536, L2 Associativity = 8



L1 Size	512	1024	2048	4096	8192
file 1	1141280	915740	769680	665300	573820
file 2	1587680	1330500	1184620	1058120	1046840
file 3	1263760	998540	730120	556180	377680
file 4	1149600	848880	581480	447540	361900
file 5	13614080	6433860	5212780	4473920	4046280
file 6	9872940	6706500	6124480	5562900	5402980
file 7	85373840	57948460	37192640	22896880	14806620
file 8	58819680	40135320	18687360	11130400	9805480

§4 Graph for Total Access Time vs L1 Associativity

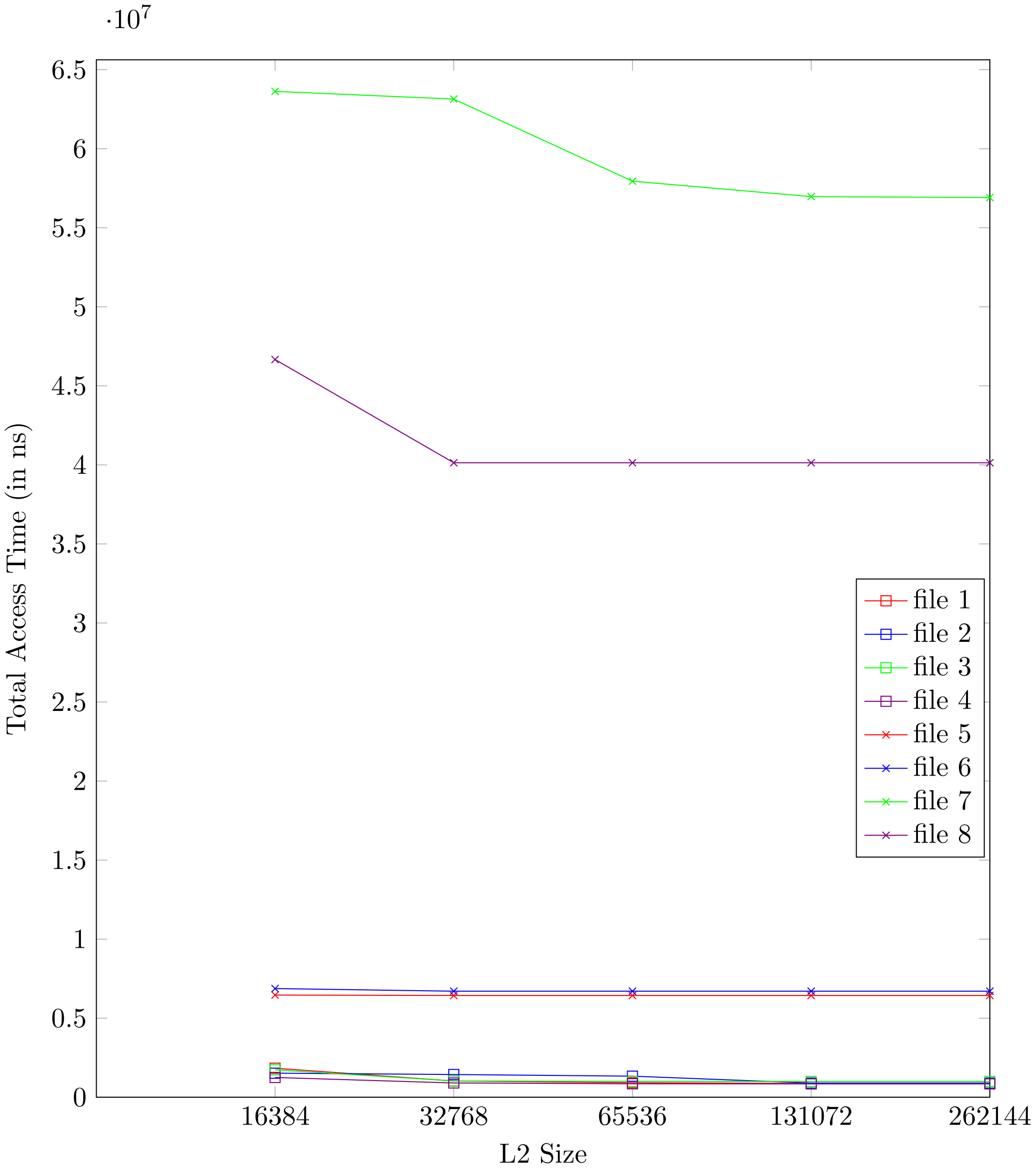
- Block size = 64, L1 size = 1024, L2 size = 65536, L2 Associativity = 8



L1 Assoc	**1**	**2**	**8**	**16**	**32**
file 1	1030580	915740	882620	855040	847140
file 2	1743240	1330500	1124600	1072060	1069140
file 3	1091940	998540	951640	937920	929960
file 4	926920	848880	751880	735640	714600
file 5	19505720	6433860	5041900	5044060	5039460
file 6	8189160	6706500	6210360	6155800	6153680
file 7	64675680	57948460	55249260	57053000	57434380
file 8	62442600	40135320	36516340	36483580	36443420

§5 Graph for Total Access Time vs L2 Size

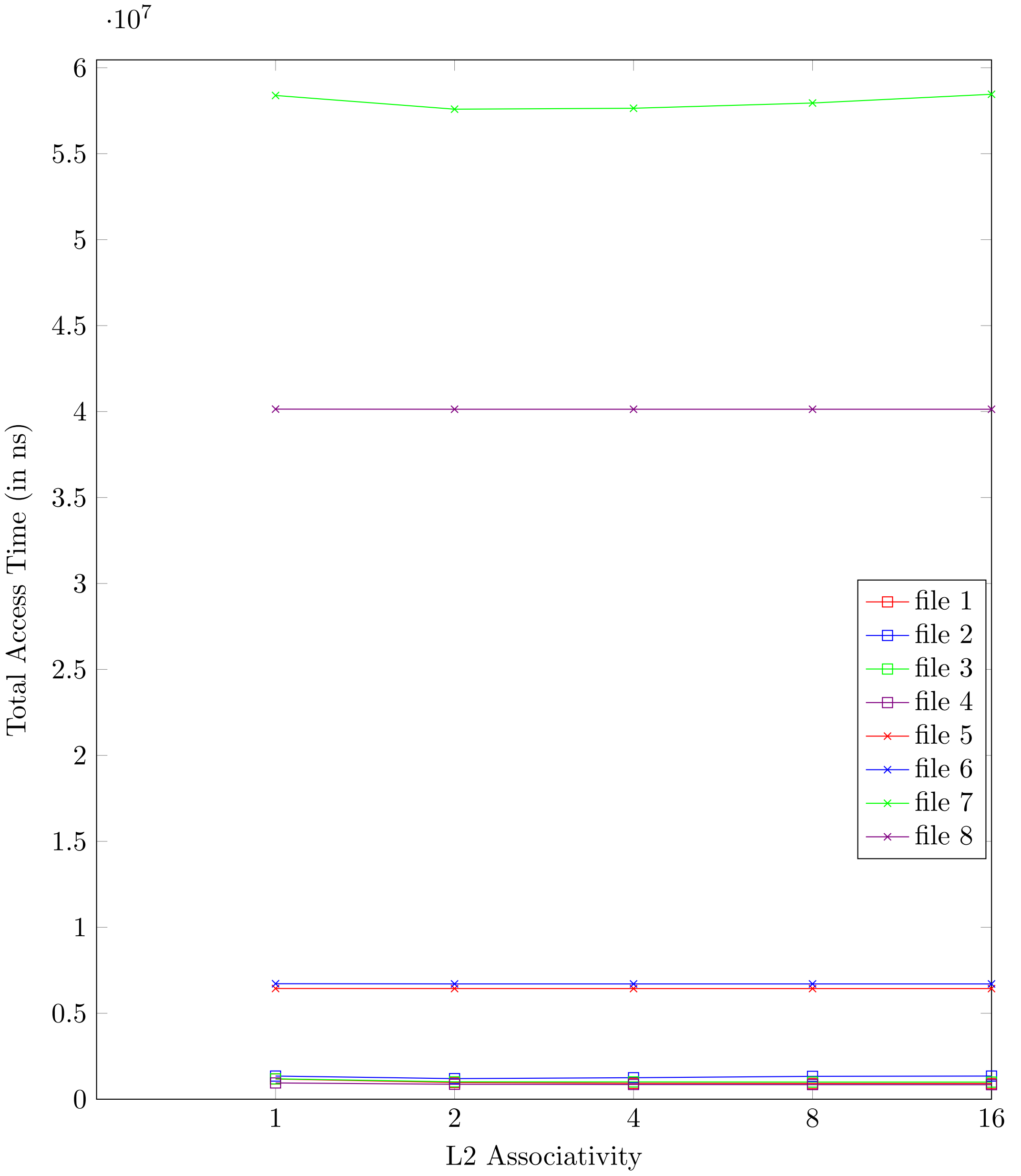
- Block size = 64, L1 size = 1024, L1 Associativity = 2, L2 Associativity = 8



L2 Size	**16384**	**32768**	**665536**	**131072**	**262144**
file 1	1834540	1025140	915740	869340	867540
file 2	1520100	1435700	1330500	895500	895300
file 3	1733540	1030140	998540	997340	997340
file 4	1247480	900080	848880	840680	840680
file 5	6460660	6433860	6433860	6433860	6433860
file 6	6872700	6706500	6706500	6706500	6706500
file 7	63628060	63143660	57948460	56971860	56910860
file 8	46666120	40135320	40135320	40135320	40135320

§6 Graph for Total Access Time vs L2 Associativity

- Block size = 64, L1 size = 1024, L1 Associativity = 2, L2 Size = 65536



L2 Assoc	**1**	**2**	**4**	**8**	**16**
file 1	1175140	970740	926940	915740	914140
file 2	1346700	1198300	1250500	1330500	1346700
file 3	1186940	1012340	1008140	998540	997340
file 4	946080	868680	855080	848880	844080
file 5	6439860	6436260	6433860	6433860	6433860
file 6	6718300	6707100	6706500	6706500	6706500
file 7	58386660	57589860	57643460	57948460	58457060
file 8	40142520	40135320	40135320	40135320	40135320

§7 Observations

- Total time decreases as we increase the size of the L1 cache by increasing the number of sets (keeping other parameters same).
- The above point does not hold for L2 cache because we are increasing the size of cache by increasing the number of sets, increasing sets decreases capacity misses but in this case there are very less capacity misses, hence total time decreases very slightly.
- As we increase the Blocksize keeping cahce size constant, initially the total time decreases as the spatial locality increases and then a dip occurs, after that total time increases again because the decrement in number of sets is more than the effect of spatial locality.
- Total time decreases as we increase the Associativity of cache, keeping size of cache constant(this is not true practically, time first decreases and then increases as we increase associativity).
- Time will definitely increase as we increase the size of trace files.

§8 Work Split

- Contribution by Raghav Ajmera : 50% , Vidushi Maheshwari : 50%